UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/669,369	09/25/2003	Horst Schnoerer	11884-406801	3373

53000 7590 06/23/2010

KENYON & KENYON LLP
1500 K STREET N.W.
WASHINGTON, DC 20005

EXAMINER	
SHUMATE, PAUL W	
ART UNIT	PAPER NUMBER
3693	
MAIL DATE	DELIVERY MODE
06/23/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/669,369	SCHNOERER ET AL.	
	Examiner	Art Unit	
	PAUL SHUMATE	3693	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 February 2010*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *5-8,15,16,18 and 21-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *5-8,15,16,18 and 21-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Status of Claims

1. This action is in reply to the communication filed on 02/22/2010.

2. Claims 1-4, 9-14, 17, 19, 20 have been canceled by Applicant.

3. Claims 5, 6, 15, 23-29 have been amended by Applicant.

4. Claims 5-8, 15, 16, 18, 21-29 have been examined and currently stand rejected.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claim(s) 5-8, 15, 16, 18, 21-29 rejected under 35 U.S.C. 103(a) as being unpatentable over Zawadzki et al., U.S. Patent No.: 7,107,268, in view of Using Microsoft Excel 97, by Hallberg, Bruce A., Sherry Kinkoph, and Bill Ray (hereinafter UME), further in view of Nakayama, U.S. Patent No. 5,317,504.

As per claims 5, 15, and 23, Zawadzki teaches a system and method for managing enterprise operations directed toward a centralized, automated, self-

maintained, collaborative project management system which manages project management objects in a hierarchical tree, comprising:

- iteratively receiving a budget item, at the computer system, for entry into the working budget database, wherein the budget item is represented by a value; (see at least column 40 lines 21-26, column 41 lines 56-60, and column 45 lines 16-18)
- executing, by a rules manager, one or more rules stored in a rules array data structure, which compare budget entries between a working budget and a reference budget, which includes a definition of a test relationship between the entries in the working budget and the entries in the reference budget and a definition of a response that is a function of the test relationship, (see at least column 3 lines 40-41, column 3 lines 45-46, column 3 lines 62-65, column 9 lines 19-24, column 10 lines 34-37, column 10 line 48, column 10 lines 56-57, column 10 line 59, column 22 lines 56-62, column 40 lines 1-51, column 41 lines 52-61, and column 65 lines 9-11)
- determining the result of the test relationship between the entry from the working budget database and the entry from the reference budget database being compared, and outputting a response defined by the response definition; (see at least column 3 lines 40-41, column 3 lines 45-46, column 3 lines 62-65, column 9 lines 19-24, column 10 lines 34-37, column 10 line 48, column 10 lines 56-57, column 10 line 59, column 22 lines 56-62, column 40 lines 1-51, column 41 lines 52-61, and column 65 lines 9-11)

- if any rule generates an error response according to the response definition, blocking the budget item from being saved to the working budget database; and otherwise, saving the received budget item in the working budget database (see at least column 10 lines 34-35, column 10 lines 55-68, and column 41 lines 52-60)
- identifying elements within the working budget database that are to be changed by the new budget item, (see at least Figure 2C, column 4 lines 42-47, column 23 lines 8-10, and column 25 lines 15-24)
- identifying rules for which the identified elements are operands, (see at least Figure 2C, column 9 lines 19-24, column 10 lines 40-45, and column 25 lines 15-24)
- wherein the executing causes only the identified rules to be executed. (see at least Figure 2C, column 9 lines 19-24, column 10 lines 40-45, and column 25 lines 15-24)

More specifically, Zawadzki teaches a rule processor and a compatibility engine which read a set of rules defined by an industry expert (see at least column 3 lines 40-41, column 3 lines 45-46, column 9 lines 19-24, and column 11 lines 16-18) and apply them against a first (source) project management object (see at least column 10 line 48) and a second (target) project management object (see at least column 10 line 59) where typical project management objects include, inter alia, organizational entities such as projects, budgets, tasks, costs, timesheets, and specs (see at least column 3 lines 62-65, column 22 lines 56-62, and column 65 lines 9-11). Zawadzki further

provides examples of comparing overall budgets, allocation budgets, and actual cost budgets to determine responses regarding whether or not a specific entry can be accepted into a budget as valid or not and how much a specific area is over or under budget (see at least column 10 lines 34-37, column 10 lines 56-57, column 40 lines 7-9, and column 41 lines 52-61). Zawadzki also teaches building a question/rule list, defining responses to the questions/rules, and applying such questions/rules to relevant components arranged in an hierarchal tree structure (see at least Figure 2C, column 10 lines 12-13, and column 12 lines 4-62), determining which rules from the set/list of rules are applicable to apply to the objects in the tree structure (see at least column 9 lines 19-24 and column 10 lines 39-45), and then applying the appropriate rules to relevant components (see at least Fig 2C).

While Zawadzki does disclose using pointers (see at least column 14 lines 21-23), test relationships (see at least column 40 lines 8-9, column 40 lines 34-36, and column 41 lines 43-52), and defined responses which depend on test relationship results (see at least column 40 lines 42-47, column 41 lines 11-21, and column 41 lines 56-58), Zawadzki does not explicitly teach that the rules themselves *include pointers* to entries within the working and reference budgets.

UME, however, teaches conditional rules used in analyzing budget items where the rules include *pointers* to both working and reference budget items, a definition of a test relationship, and a definition of a response to be made when the test relationship is not satisfied (see at least UME p. 204, paragraph(s) under IF, pp. 460-465,

paragraph(s) under Validating User Input, and p. 216, paragraph(s) under Conditional Sum Wizard).

It would have been obvious at the time the invention was made to a person having ordinary skill in the art to combine the teachings of Zawadzki and UME to form a budget management system which executes one or more rules on data, pointed to by a pointer, where the rule includes a conditional test and defined responses, which depend on the test results, because the use of pointers to reference database entries is old and well known and because the use of pointers helps avoid storing information twice in two places (see at least column 14 lines 21-26).

Regarding the limitation and arguments about the working budget database, the reference budget database, and the rules array all being stored separately, the examiner does not interpret specifying that the databases are stored separately to significantly distinguish the claims from the prior art. First, the examiner points to Applicant's own specification. In the second paragraph under Detailed Description, Applicant states that "reference to 'databases' merely connotes logically separate areas of a storage system; it is immaterial, for example, whether the working and reference budgets are provided in physically separate database systems or are merely different portions of a single database system." Further, Webster's II Dictionary, 3rd ed., defines database as "a collection of data arranged for ease or search and retrieval." Any database could be considered to be stored separately from other databases regardless if it is physically stored miles away from other databases or if it is stored in a single column in an excel sheet with another database stored one column over. Even the

databases that are side by side in an excel sheet are still in two logically different locations, even if the two logically different locations are part of a bigger single database. Therefore based on Applicant's disclosure and the explained interpretation, the examiner believes that both Zawadzki and UME sufficiently teach databases stored in at least logically different locations since the reference budget, the target budget, and the rules list are all individual entities that interact with each other.

However, it is true that neither Zawadzki or UME explicitly recite the exact words that the reference budget, the target budget, and the rules list are stored in *separate data storage areas*. Nakayama, in art very similar to the teachings of Zawadzki, teaches a database module db is constituted by records making up an item dictionary which stores data processed as needed by a command module to analyze/compare/process two or more other individual modules (see at least column 13 lines 20-24, column 14 lines 44-47, and column 14 lines 63-68). In Nakayama, many various independent modules can be applied to each other to then be applied as a whole to other modules or databases (see at least column 14 lines 44-48). It would have been obvious to one of ordinary skill in the art at the time the invention was made to specifically store various objects and database in their own modules that are separate but can be applied to each other in various ways to create and execute different functionalities because this permits anyone with little knowledge of computer systems to easily create and execute one program after another as needed. Because the commands function in the order in which they are arranged, the complexity

associated with conventional loop control arrangements is significantly alleviated (see at least column 18 lines 39-44).

As per claims 6,7, Zawadzki, in at least column 25 lines 15-24, column 38 lines 36-48, column 41 lines 11-21, column 41 lines 52-60, and column 43 lines 45-56, teaches:

- pursuant to execution of a rule, performing aggregation of addressed entries of the working database according to a definition provided in the rule, an aggregate value obtained therefrom being used to determine if the test relationship is satisfied.
- pursuant to execution of a rule, performing aggregation of addressed entries of the reference database, according to a definition provided in the rule, an aggregate value obtained therefrom being used to determine if the test relationship is satisfied.

In addition to the teachings of Zawadzki, as cited above, teachings relevant to these limitations can be found in UME on at least page 203, paragraph(s) under COUNT, COUNTBLANK, AND COUNTIF, page 208, paragraph(s) under SUM & SUMIF, and page 216, paragraph(s) under Conditional Sum Wizard.

As per claims 8 and 16, UME further teaches:

- if any rule generates a warning, posting an alert as specified in the response definition of the corresponding rule. (see at least UME p. 463-465, paragraph(s) under Setting Error Alerts and FIG. 19.13)

Zawadzki further teaches:

As per claim 18, UME, in at least p. 204, paragraph(s) under IF, pp. 460-465, paragraph(s) under Validating User Input, and p. 216, paragraph(s) under Conditional Sum Wizard, teaches:

- identifying, by using an address field, locations from a first and second budget database from which budget value information is to be obtained (UME p.204 see "C10" and "D10")
- storing in a test field a definition of a relationship that must be met between values from the first data structure and values from the second data structure to satisfy the rule (UME p.204 see "C10>D10")
- storing in a response field a definition of an action to occur if the relationship is not satisfied (UME p.204 see "Overspent")

As per claim 21, Zawadzki teaches applying a rule recursively across a plurality of sets of locations (see at least the "financial rollup component" in column 25 lines 15-25)

As per claim 22, UME teaches accessing a field for definition of an aggregation rule contained in at least one rule to the locations specified in the respective address field (see at least page(s) 410, 609, and 876)

As per claims 24-29, Zawadzki teaches a rule being applied to a single object or tree node and then because that node pointed to and depended on at least one other node, the rule was then applied to at least one more associated/affected node. (see at least column 25 lines 5-24, column 41 lines 36-37, and column 41 lines 52-60)

Response to Arguments

Applicant's arguments filed 01/25/2010 have been fully considered but they are not persuasive. Applicant argues that none of Zawadzki, UME, Nakayama, nor Zawadzki in view of UME and Nakayama fail to teach the limitations of "identifying elements within the working budget database that are to be changed by the received budget item, if the new budget item is posted into the working budget database, identifying a subset of rules that apply to the identified elements; applying only the test relationship in the subset of rules to the retrieved data and the received budget item." The examiner respectfully disagrees.

The examiner asserts that Zawadzki teaches "identifying elements within the working budget database that are to be changed by the received budget item, if the new budget item is posted into the working budget database" in at least column 4 lines 38-47, column 23 lines 8-10, and column 25 lines 15-24. Specifically, Zawadzki teaches that "the project management system of the present invention is used in the daily operations of an enterprise. In this fashion, the system is self-maintaining in that project schedules can be automatically updated as tasks are completed. Further, tasks that involve financial objects are automatically linked to other objects that are affected by them.

For example, if a purchase order object is added to a project tree, the actual and projected budgetary items associated with the current project are automatically updated to reflect the expenditure (see at least column 4 lines 38-47)." Zawadzki further teaches "project budgets are automatically updated in response to entering a purchase order

(see at least column 23 lines 8-10)" and "the final component depicted in FIG. 14 is the financial rollup component 1414. The financial rollup component is used whenever a monetary figure is added, modified or deleted to the project tree 1414. When any of these events occur, the financial rollup component 1414 updates other financially related nodes that are affected. This is accomplished by traversing the project tree 1413 in the upward direction and updating any nodes that are defined as financial, such as actual and budget dollar amounts associated with a project (see at least column 25 lines 15-25).

The examiner interprets these teachings to teach or at least strongly suggest "identifying elements within the working budget database that are to be changed by the received budget item, if the new budget item is posted into the working budget database." In Zawadzki, when a budget item/value is added, modified or deleted in the project tree, the rollup component automatically updates financially related nodes. That means the nodes financially related to where the budget item/value is added, modified or deleted, must be identified by the rollup component as nodes that may change in relation to the changed budget value. Then of course, if the identified nodes are changed in response to the added, modified or deleted budget item, then the rollup component then has to identify any nodes financially related to the updated nodes in order to automatically update the related nodes as well. This process will continue to "roll up" the tree until all relevant nodes have been identified and updated appropriately.

Regarding "identifying a subset of rules that apply to the identified elements; applying only the test relationship in the subset of rules to the retrieved data and the

received budget item." In Zawadzki, as the financial rollup component propogates up the project tree, nodes that report for a certain part of the project management tree as being over or under budget must have their select rules applied and analyzed in response to being updated to determine if the new budget item put their part of the project management tree over budget. Further, in column 9, lines 19-24, and throughout column ten, Zawadzki discloses selecting and applying appropriate subsets of rules to appropriate subsets of project tree nodes. While many of the examples are drawn towards non-budget nodes, it is obvious from reading through the reference as a whole that identifying groups of nodes in general and appropriately selecting and applying subsets of rules to be applied to certain identified nodes, as taught in various parts of Zawadzki is meant to be easily extendable to all types of project management objects (including budget objects) represented in the nodes of a project management tree.

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Paul Shumate whose telephone number is 571-270-1830. The examiner can normally be reached on M-F 8:30 AM - 6:00 PM, EST alt Fridays off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, James Kramer can be reached on 571-272-6783. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Name: Paul W. Shumate
Title: Patent Examiner
Date: 6/19/2010
Signature: /Paul Shumate/
Examiner, Art Unit 3693

/James A. Kramer/
Supervisory Patent Examiner, Art Unit 3693